

1 **CLAIMS**

2
3 1. One or more computer readable media having stored thereon a
4 plurality of instructions that, when executed by one or more processors, causes the
5 one or more processors to determine whether an input value matches any of a
6 plurality of target values by performing acts including:

7 generating a hash key based on the input value;

8 separating the hash key into a plurality of portions;

9 indexing into each of a plurality of sub-hashes using one of the plurality of
10 portions;

11 identifying a plurality of values from the plurality of sub-hashes based on
12 the indexing;

13 combining the plurality of values to generate a hash result, wherein each bit
14 in the hash result corresponds to one of the plurality of target values; and

15 for each bit in the hash result that is set, comparing the input value to the
16 corresponding target value to determine whether the values match.

17
18 2. One or more computer readable media as recited in claim 1, wherein
19 the number of target values in the plurality of target values is equal to the number
20 of bits in the hash result.

21
22 3. One or more computer readable media as recited in claim 1, wherein
23 a maximum number of target values in the plurality of target values is equal to the
24 number of bits in the result value.
25

1 4. One or more computer readable media as recited in claim 1, wherein
2 a maximum number of target values in the plurality of target values is equal to the
3 number of bits in each of a plurality of locations of the plurality of sub-hashes that
4 can be indexed.

5
6 5. One or more computer readable media as recited in claim 1, wherein
7 the separating comprises separating the hash key into two portions.

8
9 6. One or more computer readable media as recited in claim 1, wherein
10 the separating comprises separating the hash key into a plurality of contiguous and
11 equal portions.

12
13 7. One or more computer readable media as recited in claim 1, wherein
14 the combining comprises performing a bitwise logical ANDing of the plurality of
15 values.

16
17 8. A hashing architecture comprising:
18 a plurality of sub-hashes;
19 a plurality of sub-hash indexes, each index being generated from a hash key
20 and used to index into one of the plurality of sub-hashes; and
21 a combiner coupled to receive values from the plurality of sub-hashes based
22 on the plurality of sub-hash indexes, and to generate a hash result by combining
23 the received values.
24
25

1 9. A hashing architecture as recited in claim 8, wherein the combiner
2 comprises a combinatorial logic component to perform a bitwise logical ANDing
3 of the values received from the plurality of sub-hashes.

4
5 10. A hashing architecture as recited in claim 8, wherein the hashing
6 architecture is implemented in software.

7
8 11. A hashing architecture as recited in claim 8, wherein the hashing
9 architecture is implemented in firmware.

10
11 12. A hashing architecture as recited in claim 8, wherein the hashing
12 architecture is implemented in hardware.

13
14 13. A hashing architecture as recited in claim 8, wherein the plurality of
15 sub-hash indexes are generated by separating the hash key into a plurality of equal
16 portions.

17
18 14. A method comprising:
19 generating a plurality of sub-hash keys based on a hash key;
20 identifying a plurality of values from a plurality of sub-hashes by indexing
21 into each of the plurality of sub-hashes using one of the plurality of sub-hash keys;
22 and
23 generating a hash result based on the plurality of values.
24
25

1 15. A method as recited in claim 14, further comprising generating the
2 hash key prior to generating the plurality of sub-hash keys.

3
4 16. A method as recited in claim 14, wherein the generating the plurality
5 of sub-hash keys comprises separating the hash key into a plurality of equal
6 portions.

7
8 17. A method as recited in claim 14, wherein the generating the hash
9 result comprises performing a bit-by-bit logical ANDing of the plurality of values.

10
11 18. One or more computer readable media including a computer
12 program that is executable by a processor to perform the method recited in claim
13 14.

14
15 19. One or more computer readable media having stored thereon a
16 plurality of instructions that, when executed by one or more processors, determine
17 whether a security identifier of an access control element matches any of a
18 plurality of security identifiers of a security token by causing the one or more
19 processors to perform acts including:

20 generating a hash key based on the access control element security
21 identifier;

22 separating the hash key into a first portion and a second portion;

23 indexing into a first sub-hash using the first portion to identify a first sub-
24 hash value;

1 indexing into a second sub-hash using the second portion to identify a
2 second sub-hash value;

3 combining the first sub-hash value and the second sub-hash value to
4 generate a result value, wherein each bit in the result value corresponds to one of
5 the plurality of security token security identifiers; and

6 for each bit in the result value that is set, comparing the access control
7 element security identifier to the corresponding security token security identifier.

8
9 **20.** One or more computer readable media as recited in claim 19,
10 wherein the generating comprises generating the hash key by selecting a portion of
11 the access control element security identifier.

12
13 **21.** One or more computer readable media as recited in claim 19,
14 wherein the separating comprises separating the hash key into two portions that
15 include an equal number of bits and that are contiguous.

16
17 **22.** One or more computer readable media as recited in claim 19,
18 wherein the combining comprises bitwise ANDing together the first sub-hash
19 value and the second sub-hash value.

20
21 **23.** A method of determining whether an input security identifier
22 matches one or more of a plurality of target security identifiers, the method
23 comprising:

24 generating a plurality of sub-hash indexes based on a hash key;
25

1 indexing into each of a plurality of sub-hashes using a respective one of the
2 plurality of sub-hash indexes;

3 generating a result hash value by combining the plurality of values resulting
4 from indexing into the plurality of sub-hashes, wherein each of the plurality of
5 target security identifiers corresponds to a portion of the result hash value; and

6 comparing the input security identifier to at least one of the plurality of
7 target security identifiers that corresponds to a portion of the result hash value
8 having a particular value.

9
10 **24.** A method as recited in claim 23, wherein the particular value
11 comprises a value of one.

12
13 **25.** A method as recited in claim 23, wherein each portion is a bit of the
14 result hash value.

15
16 **26.** A method as recited in claim 23, wherein the generating a plurality
17 of sub-hash indexes comprises:

18 selecting a portion of the input security identifier;

19 separating the portion into two equal and contiguous sub-portions; and

20 using each of the sub-portions as one of the plurality of sub-hash indexes.

21
22 **27.** A method as recited in claim 23, wherein the generating the result
23 hash value comprises generating the result hash value by performing a bitwise
24 logical ANDing of the plurality of values.

1 28. A method as recited in claim 23, wherein the input security
2 identifier comprises an access control security identifier and each of the plurality
3 of target security identifiers comprises a security token security identifier.

4
5 29. A method as recited in claim 23, wherein the input security
6 identifier comprises a security token security identifier and each of the plurality of
7 target security identifiers comprises an access control security identifier.

8
9 30. One or more computer readable media including a computer
10 program that is executable by a processor to perform the method recited in claim
11 23.

12
13 31. A system comprising:
14 a plurality of security token security identifiers corresponding to a user;
15 a plurality of access control security identifiers corresponding to an object;
16 a plurality of sub-hashes; and
17 an access controller to determine whether any of the plurality of security
18 token security identifiers match any of the plurality of access control security
19 identifiers by, for each of the plurality of access control security identifiers,
20 generating a plurality of sub-hash indexes based on a hash key,
21 indexing into each of the plurality of sub-hashes using a respective
22 one of the plurality of sub-hash indexes,
23 identifying a plurality of values from the plurality of sub-hashes
24 based on the indexing,
25

1 combining the plurality of values to generate a hash result value,
2 wherein each bit in the hash result value corresponds to one of the plurality
3 of security token security identifiers, and

4 for each bit in the result value that is set, comparing the access
5 control security identifier to the corresponding security token security
6 identifier to determine whether the values match.

7
8 **32.** A system as recited in claim 31, wherein the plurality of sub-hashes
9 comprise two sub-hashes.

10
11 **33.** A system as recited in claim 31, wherein each bit in the result value
12 that is set has a value of one.

13
14 **34.** A system as recited in claim 31, wherein the combining comprises
15 bitwise logically ANDing together the plurality of values.

16
17 **35.** A system as recited in claim 31, wherein each of the plurality of
18 values and the hash result value each includes a number of bits equal to a
19 maximum number of security token security identifiers that can be included in the
20 plurality of security token security identifiers.

21
22 **36.** A system as recited in claim 31, wherein the system comprises an
23 operating system.
24
25

1 **37.** A system as recited in claim 31, wherein the system comprises a
2 resource manager that is not a part of an operating system.

3
4 **38.** A method comprising:
5 for each sub-hash in a plurality of sub-hashes that can be used together to
6 generate a hash result,

- 7 (a) identifying a bit in a location of a sub-hash,
8 (b) identifying, in a source value, a plurality of bits
9 corresponding to the sub-hash,
10 (c) comparing an identifier of the location to the plurality of bits,
11 (d) setting the bit if the identifier of the location matches the
12 plurality of bits, and otherwise clearing the bit, and
13 (e) repeating acts (a), (b), (c), and (d) for each of a plurality of
14 bits in the location of the sub-hash.

15
16 **39.** A method as recited in claim 38, wherein the plurality of bits
17 correspond to part of a portion of the source value that will be used to generate a
18 hash value.

19
20 **40.** One or more computer readable media including a computer
21 program that is executable by a processor to perform the method recited in claim
22 38.